

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jaka Konda

**Prepoznavanje starosti oseb s slik
obrazov z uporabo konvolucijskih
nevronskih mrež**

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Peter Peer

Ljubljana, 2016

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Prepoznavanje starosti oseb s slik obrazov z uporabo konvolucijskih nevronskih mrež

Tematika naloge:

Preučite področje prepoznave starosti oseb s slik obrazov, pri čemer se osredotočite predvsem na konvolucijske nevronske mreže. Zasnуйте ustrezno arhitekturo mreže za ta problem, uporabite primerne slikovne zbirke za učenje in testiranje ter primerjajte dobljene rezultate z rešitvami na tem področju.

*Zahvaljujem se staršem, ki sta me ves čas podpirala in mi stala ob strani.
Zahvaljujem se tudi mentorju izr. prof. dr. Petru Peeru za strokovno svetovanje in mentorstvo pri izdelavi diplomskega dela.*

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Nevronske mreže	3
2.1	Delovanje in zgradba	4
2.2	Učenje nevronske mreže	7
2.3	Vrste nivojev	11
3	Podatkovne zbirke	15
3.1	Podatkovna zbirka IMDB-WIKI	15
3.2	Podatkovna zbirka LAP 2015	17
4	Predlagana metoda	19
4.1	Priprava podatkov	19
4.2	Knjižnica Caffe za delo s konvolucijskimi nevronske mrežami	20
4.3	Model konvolucijske nevronske mreže	22
4.4	Učenje nevronske mreže	24
5	Rezultati	27
5.1	Rezultati za podatkovno zbirko LAP 2015	27
5.2	Primerjava s sorodnimi rešitvami	29
6	Zaključek	33
	Literatura	39

Seznam uporabljenih kratic

kratica	angleško	slovensko
AFLW	Annotated Facial Landmarks in the Wild	označene obrazne značilke v nekontroliranem okolju
CACD	Cross-Age Celebrity Dataset	starost znanih skozi čas
CNN	Convolutional Neural Network	konvolucijska nevronska mreža
DPM	Deformable Parts Model	sestavljene modeli
NPD	Normalized Pixel Difference	normalizirana razlika slikovnih elementov
HOG	Histogram of Gradients	histogram gradientov
ICCV	International Conference on Computer Vision	mednarodna konferenca o računalniškem vidu
ILSVRC	ImageNet Large Scale Visual Recognition Challenge	izziv prepoznavne objektov s slik
JSON	JavaScript Object Notation	notacija objektov JavaScript
LAP	Looking at People	gledanje ljudi
LDA	Linear Discriminant Analysis	linearna diskriminantna analiza
PCA	Principal Component Analysis	analiza glavnih komponent
ReLU	Rectified Linear Units	usmerjene linearne enote
SVM	Support Vector Machine	metoda podpornih vektorjev
VGG	Visual Geometry Group	skupina za vizualno geometrijo

Povzetek

Naslov: Prepoznavanje starosti oseb s slik obrazov z uporabo konvolucijskih nevronske mreže

Diplomsko delo pokaže celoten postopek razvoja rešitve problema prepoznavanja starosti oseb s slik. Začnemo s teoretičnimi osnovami o konvolucijskih nevronske mrežah, s pomočjo katerih smo se tudi lotili problema. V praktičnem delu sledi priprava podatkov in učenje sestavljenega modela nevronske mreže, kjer smo izbrali znano VGG arhitekturo. Naučen model preizkusimo še na tekmovanju LAP, da dobimo rezultate, ki jih nato primerjamo z rešitvami ostalih ekip. Naši rezultati so se kljub nekoliko preprostejšemu pristopu izkazali za precej vzpodbudne, saj smo dosegli manjšo napako kot človek ter se uvrstili na 4. mesto med 11 ekipami.

Ključne besede: računalniški vid, strojno učenje, nevronska mreža, konvolucijska nevronska mreža, klasifikacija, zaznavanje obrazov, starost, klasifikacija starosti, določanje starosti.

Abstract

Title: Recognising people's age from face images with convolutional neural networks

The diploma thesis presents the entire process of developing a solution for recognising person's age in the image. We start with the theoretical basics of convolutional neural networks that we used to address the problem. In the practical part we start with the preparation of used datasets and continue with learning of our neural network with the chosen widely known VGG architecture. Learned model is tested on the LAP competition dataset in order to obtain results, which are comparable with the solutions of other teams. Despite somewhat simpler approach our results proved to be quite encouraging. We surpassed human performance and ranked 4th among 11 teams.

Keywords: computer vision, machine learning, neural networks, convolutional neural networks, classification, face detection, age, age classification, age recognition.

Poglavje 1

Uvod

Računalniški vid je široko področje, ki se ukvarja z razumevanjem informacije, podane v obliki slikovnega gradiva. Združuje več različnih področij, predvsem matematiko, računalništvo, strojno učenje in obdelavo signalov, s ciljem gradnje boljših modelov in algoritmov za prepoznavanje različnih objektov na slikah, klasifikacijo, gradnjo opisnikov ter značilk. Dandanes se široko uporablja v najrazličnejših panogah, od potrošniških naprav, kjer se vsakdanje običajno srečujemo z zaznavanjem obrazov pri fotoaparatih, obdelave slik, vedno bolj popularne navidezne resničnosti, biometrike pri zagotavljanju dostopa glede na prstni odtis ali šarenice posamezne osebe, do industrijskih aplikacij in robotike, kot je avtonomna vožnja, nadzorovanje proizvodnih procesov in zagotavljanje kvalitete.

Eden izmed odprtih problemov je določanje starosti osebe zgolj na podlagi informacije, pridobljene s slike obraza. Tvrstna informacija je uporabna v mnogo aplikacijah, kot so prilagoditev sistema glede na starostno skupino uporabnika, pridobivanje podatkov o demografiji uporabnikov v trgovskih centrih, omejevanje dostopa glede na starost uporabnika, verifikacija vnosenih podatkov osebe v raznih obrazcih in na socialnih omrežjih ali zgolj kot značilka v pomoč pri identifikaciji osebe. V namen reševanja tovrstnega problema vsako leto poteka tekmovanje LAP [1], kjer je cilj napovedati starost osebe na sliki.

Tako kot ljudje tudi uspešnejši računalniški sistemi pri določanju očitne starosti upoštevajo zunanje značilnosti obraza, ki se kažejo s staranjem. Glavne lastnosti v zgodnejših letih predstavljajo rast kosti in s tem obliko obraza, kasneje pa pojavitev gub, najpogostejše okoli oči, izgube moči obraznih mišic, od koder sledi povešenost in sprememba teksture kože [2]. Te lastnosti se pri posamezniku pojavijo v različnih časovnih obdobjih v različni meri. Zato ločimo očitno starost, kako oseba izgleda navzven in starost in koliko je posameznik dejansko star (čas od njegova rojstva). Dodatno določanje starosti otežijo tudi druge lastnosti, kot so izraz na obrazu in spol, v domeni računalniškega vida pa, poleg naštetega, sem sodijo še občutljivost na različno osvetlitev slike oz. učnih primerov, orientacija obraza in kvaliteta slike.

Na tem področju obstaja že nekaj uspešnih rešitev. Prve so združevale leta v kategorije, kot so mlad, srednjih let in star [3], kasnejše pa so leta združevala po deset skupaj v eno skupino [4]. Vse omenjene metode so se problema lotile z uporabo metod strojnega učenja, kjer prevladujejo algoritmi PCA, LDA in SVM, nekatere novejšje so uporabile tudi že nevronske mreže.

V diplomskem delu se bomo osredotočili na postopek reševanja omenjenega problema z globokimi konvolucijskimi nevronskimi mrežami s sodobno VGG [5] arhitekturo. V poglavju 2 se bomo osredotočili na delovanje nevronskih mrež, njihovo zgradbo in proces učenja. Za njihovo delovanje potrebujemo ogromne podatkovne množice, ki bodo opisane v poglavju 3. V poglavju 4 sledi priprava podatkov za učenje, opis in učenje modela, kar predstavlja glavni cilj naloge. V poglavju 5 sledijo rezultati in primerjava z obstoječimi rešitvami na znanih podatkovnih množicah, kot je LAP. V poglavju 6 je zaključek, v katerem analiziramo uporabljeni pristop in navedemo možne izboljšave.

Poglavje 2

Nevronske mreže

Možgani so se v biologiji izkazali za neverjetno prilagodljive in zmožne rešiti tudi probleme, ki jih ni mogoče formulirati kot algoritem. Delujejo na podlagi iskanja vzorcev, ravno nasprotno od računalnikov, ki so izvrstni v izvajanju algoritmov. Dandanes je vedno večja potreba ravno po avtomatizaciji problemov, kot sta prepoznava objektov s slike in govora.

Če primerjamo zgolj lastnosti možganov in računalnikov (tabela 2.1), slednji ne zaostajajo prav veliko. Po zgledovanju iz narave so se tako v računalništvu razvile umetne nevronske mreže, ki poskušajo na preprost način predstaviti delovanje in učenje v računalnikih. S tovrstnimi algoritmi so tako že na nekaj področjih dosegli in presegli rezultate, ki jih dosegajo ljudje [6].

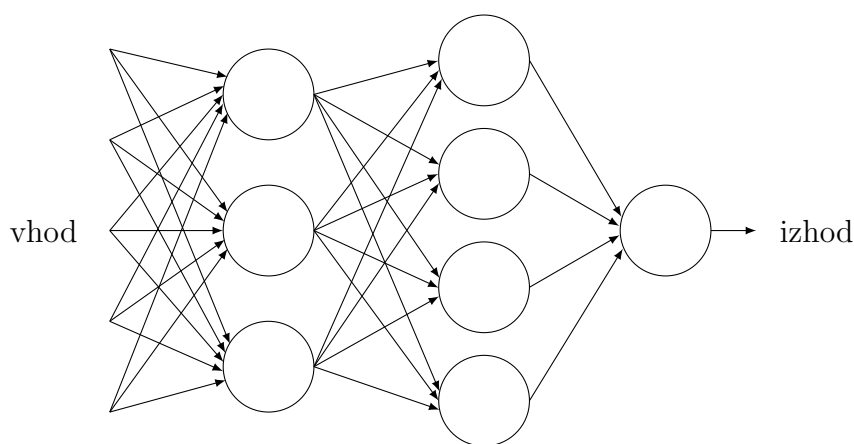
	Človeški možgani	Računalnik
Število procesnih enot	$\approx 10^{11}$	$\approx 10^9$
Tip procesnih enot	Nevroni	Tranzistorji
Način izračunov	Vzporedno	Običajno serijsko
Preklopni čas	$\approx 10^{-3}\text{s}$	10^{-9}s
Teoretično možnih prekopov	$\approx 10^{13}/\text{s}$	$10^{18}/\text{s}$
Prekopov v praksi	$10^{12}/\text{s}$	$10^{10}/\text{s}$

Tabela 2.1: Primerjava lastnosti človeških možganov in računalnika [7].

2.1 Delovanje in zgradba

Podobno kot pri bioloških mrežah so tudi umetne nevronske mreže sestavljene iz različnih zaporednih nivojev, kjer je posamezni nivo sestavljen iz nevronov. Med nivoji so povezave, ki v biologiji predstavljajo aksone. Vsaka nevronska mreža mora imeti vsaj dva nivoja. Prvi nivo, ki ga ne štejemo med število nivojev nevronske mreže, predstavlja vhodni nivo. Pri vhodnem nivoju mora biti število vhodnih nevronov enako številu atributov v podatkih ali slikovnih elementov, če obdelujemo slike. Zadnji nivo pa predstavlja izhodni nivo, kjer je izhod v primeru regresije nevron oziroma, v primeru klasifikacije, večje število izhodnih nevronov, kar je enako številu vseh razredov [8]. Vrednosti si lahko predstavljamo kot verjetnosti, da primerek spada v določen razred. V tem primeru za izbiro rezultata izberemo razred, čigar pripadajoč nevron ima največjo vrednost.

Med vhodnim in izhodnim nivojem so lahko tudi skriti nivoji več vrst, nekaj ključnih je opisanih v poglavju 2.3. Preprost primer nevronske mreže z dvema skritima nivojema je podan na sliki 2.1.



Slika 2.1: Primer nevronske mreže s tremi nivoji, en vhodni (ki ga ne štejemo med nivoje nevronske mreže), dva skrita in izhodni nivo.

Pri učenju se na posameznem nivoju tako ustvarijo vzorci, ki so najlepše vidni v primeru obdelave slik. V začetnih nivojih so ti vzorci preprosti, recimo črte pod različnimi koti in podobni preprosti vzorci. Na naslednjih

nivojih se iz teh preprostih vzorcev tvorijo kompleksnejši in tako naprej. Na sliki 2.2 je prikaz vzorcev, ki se jih nauči konvolucijska nevronska mreža med učenjem prepoznave obrazov [9].



Slika 2.2: Prikaz vzorcev prvega skritega nivoja v konvolucijski nevronske mreži, vmesnega in zadnjega konvolucijskega nivoja pri učenju prepoznave obrazov.

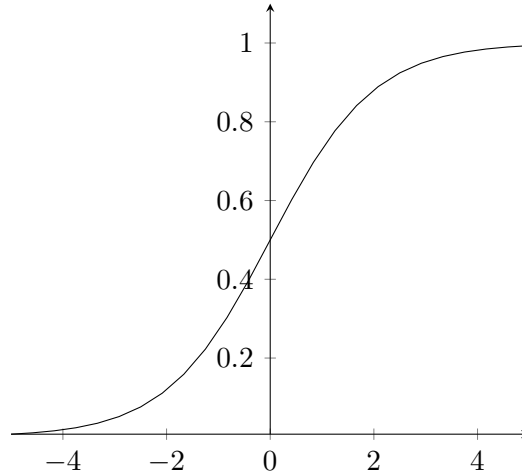
2.1.1 Nevroni

Nevroni predstavljajo najmanjšo gradbeno enoto nevronske mreže. Na začetku so to bili perceptroni [10], kjer je bil njihov izhod odvisen le od seštevek vhodov. Če je vhod večji od nekega praga, vrne 1, sicer 0. V enačbi (2.1) je poleg vhoda x_j podan še w_j , ki predstavlja težo povezave med nevronom prejšnjega nivoja in trenutnega. Utež pove pomembnost podatka, ki pride po tisti povezavi, določi se podobno kot meja v fazi učenja modela.

$$\text{izhod} = \begin{cases} 0, & \text{če je } \sum_j w_j x_j \leq \text{prag} \\ 1, & \text{če je } \sum_j w_j x_j > \text{prag} \end{cases} \quad (2.1)$$

Zaradi omejenosti izhoda perceptronov so se razvili logistični nevroni, katerih izhod lahko zavzame katerokoli vrednost na odprtem intervalu med 0 in 1. Najpogosteje uporabljen nevron tovrstne vrste je sigmoidov nevron [11], ki deluje po sigmoidovi funkciji v enačbi (2.2); prikazani na sliki 2.3. Pogosta alternativa je hiperbolični tangens. Tovrstne funkcije so znane kot aktivacijske funkcije. Teža povezav se na takšnih funkcijah odraža kot hitrost naraščanja.

$$\frac{1}{1 + e^{-x}} \quad (2.2)$$



Slika 2.3: Vizualni prikaz sigmoidove funkcije.

2.1.2 Korekcijska vrednost

V mrežah so običajno prisotne dodatne korekcijske vrednosti (angl. bias), ki so se velikokrat izkazale kot kritične pri učenju. Korekcijsko vrednost si lahko predstavljamo kot zamik vhodne funkcije levo ali desno. Služi tudi kot pomoč v primeru, da so vsi vhodi enaki 0, saj vseeno vrne nek rezultat. Korekcijska vrednost je običajno ena za celoten nivo, redkeje pa ena na posamezni nevron. Enačba (2.3) je le nadgradnja izhodne funkcije nevrona. Dodan ji je še b , ki predstavlja korekcijsko vrednost [12].

$$y = f\left(\sum_j w_j x_j + b\right) \quad (2.3)$$

2.2 Učenje nevronske mreže

Nevronske mreže se po kategorijah uči na tri različne načine. Ti so gradientni spust z vzvratnim propagiranjem napake (angl. backpropagation) [13], optimiziranje z delci roja (angl. particle swarm optimization) [14] in genetski algoritmi [15]. Način z uporabo gradientnega spusta z vzvratnim popravljanjem je najširše uporabljen, zato je opisan v nadaljevanju.

2.2.1 Gradientni spust (z momentom)

Najširše uporabljen pristop učenja nevronske mreže je zaradi preprostosti, hitrosti in zanesljivosti gradientni spust (angl. gradient descent).

Predstavlja optimizacijski algoritem, ki na izhodu minimizira napako in s tem povečuje točnost. V točki, kjer se trenutno nahajamo, se izračuna gradientni vektor, ki kaže v smeri največjega vzpona oz. spusta funkcije, če minimiziramo napako. Nato se premaknemo za majhen del tega vektorja v novo točko. Velikost dela vektorja nam pove hitrost učenja. Ta postopek ponavljamo, dokler ne dosežemo konvergence v nek lokalni optimum. Vektor gradienta se izračuna iz parcialnih odvodov funkcije f po enačbi (2.4), za vsak izhod posebej (x_i).

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \quad (2.4)$$

Gradientni spust bi lahko primerjali z vzponom na hrib, kjer želimo brez poznavanja poti čim hitreje priti na vrh. V vsakem koraku pogledamo, kje se najhitreje vzpenja in se pomaknemo korak tja. Postopek ponavljamo, dokler ne pridemo na vrh.

Težava nastane, če obtičimo na lokalnem vrhu. Kamorkoli se premaknemo pridemo v slabši položaj, vendar nismo na najvišjem. To v mnogo primerih predstavlja težavo, zato se v praksi običajno uporablja gradientni spust z momentom. Ob vsaki posodobitvi vektorja smeri upoštevamo še nek delež λ prejšnjih vektorjev smeri, kjer njihov vpliv pada eksponentno. Če je λ enaka 0, govorimo o navadnem gradientnem spustu. Tako se v primeru, ko

se približamo nekemu lokalnemu optimumu, lahko premaknemo naprej le, če je moment dovolj velik.

2.2.2 Vzratno računanje pričakovane napake

Nevronske mreže z enim nivojem so sposobne rešiti le preproste probleme, zato so običajno sestavljene iz več nivojev. S tem se zmanjša potreba po številu nevronov. Med učenjem vemo le, katere podatke pošljemo v vhodni nivo, pravilne rezultate in predvidene izhode iz zadnjega nivoja. Vse vmesne uteži moramo tako popravljati glede na vhod in izhod, kot tudi upoštevati vpliv na ostale neurone. Za to se uporablja algoritem za vzratno računanje pričakovane napake. V nadaljevanju je razložen postopek računanja; oznake so na sliki 2.4.

Legenda:

j – izhodni nevron

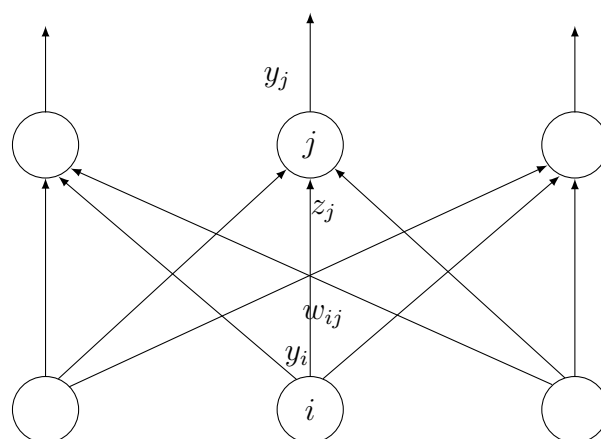
i – skriti nevron

y_j – izhod nevrona j

y_i – izhod nevrona i

z_j – skupni vhod nevrona j

w_{ij} – teža povezave iz nevrona i v j



Slika 2.4: Primer grafa nevronske mreže z oznakami.

V prvem koraku z verižnim pravilom izračunamo, kolikšno napako E povzročimo na izhodu y_j s spremembo vhoda v nevron z_j :

$$\frac{\partial E}{\partial z_j} = \frac{dy_j}{dz_j} \frac{\partial E}{\partial y_j} \quad (2.5)$$

V naslednjem koraku izračunamo spremembo napake na izhodu, če spremenimo izhod nevrona i (y_i). Upoštevati moramo napako, ki jo povzročimo na vseh nevronih na naslednjem nivoju povezanim z nevronom i . Napaka tako predstavlja seštevek napak vseh naslednjih nevronov in smo jo izračunali v prejšnjem koraku.

$$\frac{\partial E}{\partial y_i} = \sum_j w_{ij} \frac{\partial E}{\partial z_j} \quad (2.6)$$

V zadnjem koraku le še izračunamo vpliv spremembe posamezne uteži na napako. To pa predstavlja izhod nevrona skupaj s pričakovano napako.

$$\frac{\partial E}{\partial w_{ij}} = y_i \frac{\partial E}{\partial z_j} \quad (2.7)$$

Za posodobitev uteži na skritem nivoju in v nevronih tako uporabimo optimizacijsko metodo gradientnega spusta, glede na pričakovano napako.

Ta postopek lahko uporabimo za poljubno število nivojev, vendar bo s povečevanjem števila nivojev prihajalo do večjih napak. Zaradi teh bo postopek postal neučinkovit.

2.2.3 Učenje z ostanki

Pri navadnem učenju si lahko vsak nivo predstavljamo kot računanje funkcije f_i , prikazano v enačbi (2.8), kjer x predstavlja izhod in f_n izhodni nivo. Zaradi kopičenja napak pri vzratnem računanju skozi računanje vseh funkcij, postane gradientni spust neučinkovit.

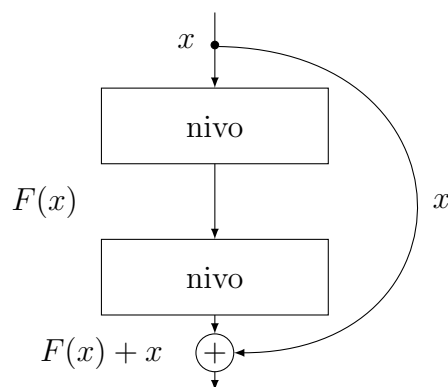
$$y = f_1(f_2(\dots(f_n(x))) \quad (2.8)$$

Da bi se temu izognili pri učenju globokih mrež, se uporablja metoda učenja z ostanki (angl. residual learning) [16], ki je le dodatek k vzratnemu računanju pričakovane napake. Mrežo se razdeli v bloke. Primer bloka z oznakami za izračun je na sliki 2.5. Nivoji znotraj bloka predstavljajo neko funkcijo $F(x)$. Predpostavimo¹, da lahko več nivojev $F(x)$ približno izračuna katerokoli funkcijo $H(x)$, ne glede na njeno zahtevnost. Tako, če od $H(x)$ odštejemo ostanek y (napaka funkcije $F(x)$), pridemo nazaj do napovedi nivojev ($F(x)$); enačba (2.9).

$$F(x) = H(x) - y \quad (2.9)$$

Če predpostavko obrnemo, lahko s funkcijo $F(x)$ približno izračunamo tudi ostanek y , za vhod pa vzamemo le x . $F(x)$ tako nadomesti y in postane razlika med vhodom x in funkcijo $H(x)$. Če obrnemo, dobimo, da je seštevek ostanka in vhoda x enak željeni funkciji; prikazano v enačbi (2.10). V praksi napaka še vedno ostaja, vendar je veliko manjša kot pri običajnem učenju.

$$\begin{aligned} x &= H(x) - F(x) \\ F(x) + x &= H(x) \end{aligned} \quad (2.10)$$



Slika 2.5: Primer dvonivojskega bloka znotraj nevronske mreže.

¹Ta predpostavka ni dokazana in je še nerešen problem [17].

S tem v mreži za vsak blok popravimo podatke, ki jih uporabljamo pri vzvratnem računanju za gradientni spust, kar nam omogoča precej globlje nevronske mreže. Avtorji so uspešno preizkusili mreže, ki imajo tudi 1201 nivojev in dosegli izvrstne rezultate. Tovrstna arhitektura mrež se imenuje ResNet.

2.3 Vrste nivojev

Nevronske mreže so, kot že omenjeno, sestavljene iz več različnih nivojev. Prvotne so imele le eno vrsto nivoja, ta je bil polno povezan. Kasneje so se razvili še dodatni nivoji, nekaj najbolj pogostih je opisanih v naslednjih podpoglavjih.

2.3.1 Polno povezani nivo

Pri polno povezanem nivoju je vsak nevron povezan z vsemi nevroni iz prejšnjega nivoja. Pri sestavljenih nevronskih mrežah se takšni nivoji običajno pojavljajo proti koncu. Predstavljajo povezave med vzorci iz prejšnjih nivojev. Prehod skozi polno povezan nivo je enak operaciji matričnega množenja.

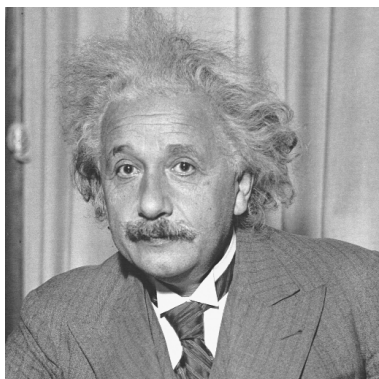
2.3.2 Konvolucijski nivo

Nad danimi vhodnimi podatki v nivo se izvede konvolucija [18]. V matematiki je konvolucija operacija med dvema matrikama, kjer je rezultat prav tako matrika. Prva matrika predstavlja podatke, medtem ko druga filter (pravimo ji tudi jedro, angl. kernel), s katerim gremo skozi podatke. Prehod si lahko predstavljamo kot drseče okno, s katerim gremo skozi vse možne lokacije v podatkih. Te uteženo seštejemo s filtrom, da dobimo vrednost in se premaknemo na naslednjo lokacijo.

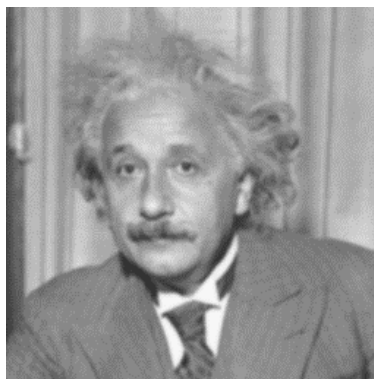
Konvolucija je v obdelavi signalov in slik izredno široko uporabljena. Eden izmed načinov vsakdanje rabe pri obdelavi slik je zameglitev slike. Filter v enačbi (2.11) je uporabljen za zameglitev slike s povprečenjem sose-

dnjih slikovnih elementov. Rezultat konvolucije slike 2.6a je tako zamegljena slika 2.6b.

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2.11)$$



(a) Primer slike pred obdelavo s konvolucijskim filtrom.



(b) Po obdelavi s konvolucijskim filtrom, kjer je slika zamegljena.

Slika 2.6: Primer obdelave slike pred in po konvoluciji.

Konvolucija je tako zaradi vseh možnih postavitvev okna računsko izredno zahtevna operacija, kjer se je potrebno v vsakem oknu sprehoditi čez vse vrednosti. Da bi se v prihodnjih nivojih izognili nepotrebnemu računanju, se uporablja optimizacijski postopek združevanje največjih (angl. max pooling) [19]. Konvuliran rezultat se razdeli v mrežo, najpogosteje tako, da so celice visoke in široke dva elementa in iz vsake celice se vzame element z največjo vrednostjo, ostale pa se zavrže. Vizualni prikaz postopka združevanja največjih je viden na sliki 2.7. Tvrstna optimizacija se zelo pogosto uporablja, kjer imamo na voljo malo podatkov, da se izognemo prekomernemu prilagajanju učnih primerov (angl. overfitting).

Rezultat konvolucijskega nivoja so tako vzorci; kot je vidno na sliki 2.2 v uvodu poglavja. Omenjeni vzorci se iščejo na vhodnih podatkih in na izhodne sporočajo verjetnosti, da se je vzorec pojavil. Pri učenju se prav tako tvorijo prostorske relacije med posameznimi vzorci.



Slika 2.7: Levo matrika pred in desno po združevanju največjih elementov.

2.3.3 Nivo usmerjenih linearnih enot

Usmerjene linearne enote (angl. rectified linear units, kratko ReLU) [20] so sodobna alternativa sigmoidnim nivojem, ki so se pri globokih nevronske mrežah izkazale celo bolje. So mnogo hitrejše in preprostejše, njihova funkcija pa je, da vse negativne vrednosti zamenja z 0, pozitivne pa obdrži (enačba 2.12).

$$f(x) = \max(0, x) \quad (2.12)$$

2.3.4 Izgubni nivoji

Izgubni nivoji se pojavljajo kot zadnji nivo. Računajo hitrost gradientnega spusta glede na pravilno in napačno napovedano učne primere. Pri klasifikaciji je najpogostejši normalizacijski nivo (angl. softmax) [21], pri regresiji pa evklidska razdalja do pravilnega rezultata. V primeru pravilno napovedanih rezultatov bo korak pri učenju z gradientnim spustom zelo majhen ali ga pa ne bo, če je pa napak zelo veliko, bodo sorazmerno veliki tudi popravki.

Poglavje 3

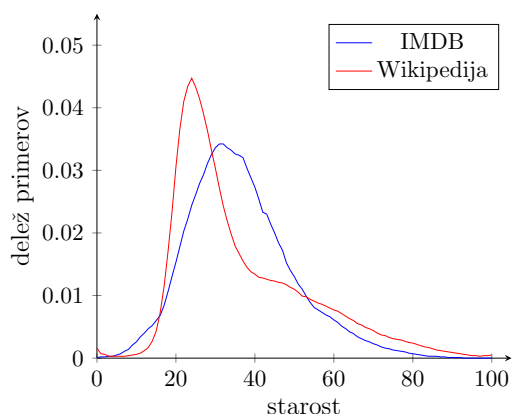
Podatkovne zbirke

Naučeni modeli so običajno tako dobri kot učni podatki. Predvsem je pomembna raznolikost podatkov in čim večje število učnih primerov, saj se model tako nauči variacij znotraj podatkov in postane na njih neobčutljiv. Primeri variacij so pogled obraza (stranski ali frontalni), neporavnano obraz na sliki in izraz na obrazu. V tem poglavju bomo opisali dve uporabljeni podatkovni zbirki, ki ju bomo uporabili za učenje. Prva je IMDB-WIKI [22] in druga LAP 2015 [23], ki se pojavlja na tekmovanjih za primerjavo različnih metod.

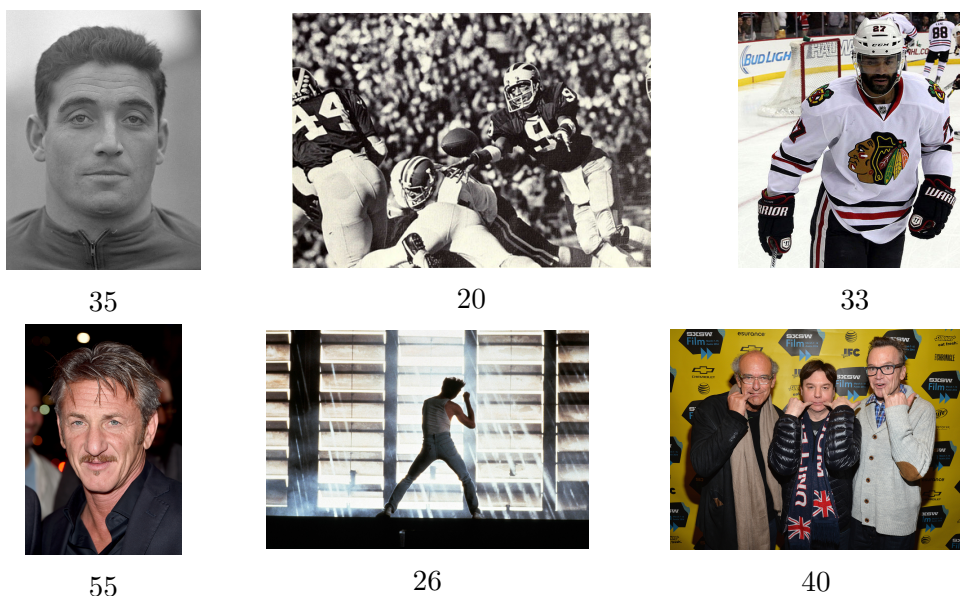
3.1 Podatkovna zbirka IMDB-WIKI

Zbirka IMDB-WIKI je trenutno ena izmed največjih zbirk na svetu, ki vključuje fotografije oseb z označeno starostjo ter spolom. Podatki so bili zbrani s spletne strani IMDB in Wikipedija. Prva je spletna stran o filmih in igralcih, druga pa spletna enciklopedija, ki vsebuje tudi biografije znanih oseb. S prvega vira so bili prebrani ime, priimek, spol in datum rojstva 100.000 najbolj znanih oseb s slikami, ki vsebujejo datum posnetka. Podobno je bilo storjeno na Wikipediji, od koder so bile vzete biografije oseb in pripadajoče slike. Skupaj je bilo tako zbranih 523.051 slik, od tega 460.723 iz IMDB in 62.328 z Wikipedije. Razporeditev podatkov po starosti je prikazana na

sliki 3.1. Skupna velikost podatkovne zbirke znaša 282 GB. Na precejšnjem številu slik je vidnih več oseb, obraz ni glavni del slike, ali pa sploh ni viden. Nekaj primerov je vidnih na sliki 3.2. Za uporabo zbirke torej potrebujemo še predhodno obdelavo podatkov.



Slika 3.1: Graf normalizirane razporeditve primerov v zbirkah IMDB in Wikipedija po starosti.



Slika 3.2: V prvi vrsti primeri treh fotografij z Wikipedije in spodaj pripadajoče starosti, v drugi s spletne strani IMDB.

3.2 Podatkovna zbirka LAP 2015

Podatkovna zbirka LAP je del izziva Napovedovanje starosti posameznikov na slikah, ki je bil del ICCV delavnice [24]. Vključuje 5.000 fotografij posameznikov z oznako starosti. Zbirka je že s strani organizatorjev razdeljena na tri dele: učni, testni in ocenjevalni, s približno razporeditvijo 50 %, 25 % in 25 % vseh fotografij, vendar nam ocenjevalni del ni na voljo. Oznake so bile zbrane s pomočjo dveh spletnih aplikacij. Vsako fotografijo je ocenilo najmanj deset ljudi, povprečje ocen pa predstavlja oznako vsake fotografije. Za vsako fotografijo je podan tudi standardni odklon ocen glasov, ki se ga uporabi pri evaluaciji rezultatov. Ta je med 1 in 6,7 s povprečjem 4. Na vsaki fotografiji je le en obraz, vendar je podobno kot pri zbirki IMDB-WIKI tudi tu potrebna predhodna obdelava fotografij. Nekaj primerov fotografij je prikazanih na sliki 3.3. Razporeditev podatkov je zelo podobna kot z Wikipedije na sliki 3.1.



$$\mu = 11,0 \quad \sigma = 2,60$$



$$\mu = 39,0 \quad \sigma = 5,30$$



$$\mu = 20,2 \quad \sigma = 4,0$$



$$\mu = 60,9 \quad \sigma = 4,9$$

Slika 3.3: Nekaj primerov fotografij iz zbirke LAP 2015 in pripadajoče povprečne starosti (μ) ter standardni odklon ocen (σ).

Poglavje 4

Predlagana metoda

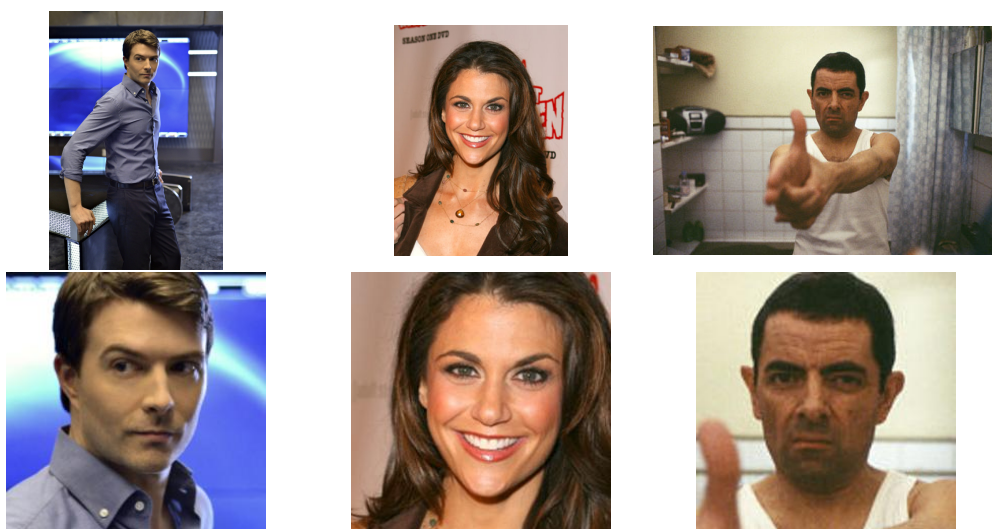
4.1 Priprava podatkov

V podatkovni zbirki IMDB-WIKI so dane neobdelane slike, zbrane s spletnimi pajki. Nekatere vsebujejo napačne časovne žige (čas zajema slike). Tako smo naleteli na skrajne primere, kjer je bila oznaka starosti osebe na sliki negativna ali pa je bila oseba stara več tisoč let. Takšne primere smo izločili v prvem koraku, saj so seveda nepravilni.

Druga težava slik v podatkovni zbirki je, da so poleg obraza vidni še ostali deli telesa in okolica, na kateri je lahko več oseb. V prvem koraku smo zato zaznali obraze na slikah z metodo NPD [25]. Uporabili smo že naučen model za zaznavo obrazov, dan s strani avtorjev algoritma. Model je bil naučen na zbirki AFLW [26], ki skupaj vsebuje nekoliko manj kot 26.000 slik z označenimi lokacijami obrazov pod različnimi koti. Obdržali smo vse slike, kjer je bil zaznan le en obraz. Vse regije zaznanih obrazov smo povečali za 60 %, izrezali le ta del ter zmanjšali ali povečali sliko na velikost 256×256 slikovnih pik. Primeri izrezanih slik so na sliki 4.1. Skupno nam je tako ostalo 175.457 slik za učenje. Regijo obreza obrazov (povečavo za 60 %) smo določili eksperimentalno, tako da so vse opazovane slike zajemale celoten obraz.

Podobno smo naredili tudi na podatkovni zbirki LAP. Na sliki smo iz-

rezali regijo, kjer je detektor vrnil najvišjo oceno za možnost obraza, saj že organizatorji zagotovijo, da je na vsaki sliki le en. Vse ostale regije so bile zavrnjene.



Slika 4.1: V prvi vrsti primeri slik iz podatkovne zbirke IMDB-WIKI pred obdelavo in v drugi vrsti po njej.

4.2 Knjižnica Caffe za delo s konvolucijskimi nevronskimi mrežami

Knjižnica Caffe [27] je ena izmed najbolj uporabljanih knjižnic (če ne kar najbolj) za delo s konvolucijskimi nevronskimi mrežami. Razvijajo jo na univerzi Berkley. Glavni aduti knjižnice so preprostost, modularnost in hitrost zaradi uporabe grafičnih kartic. Modularnost jo naredi privlačno za mnogo raziskovalcev, saj se jo zelo hitro dograjuje z najnovejšimi dognanji na področju nevronskih mrež. Z uporabo grafičnih kartic pa se razbremeni nekaj procesorskih jeder (običajno od 4 do 8) in tako se delo prenese na nekaj tisoč grafičnih jeder. S tem se čas učenja zmanjša iz nekaj mesecev na nekaj dni. Caffe podpira Nvidiine grafične kartice s tehnologijo CUDA, za še hitrejše izvajanje pa uporablja knjižnico cuBLAS (angl. basic linear algebra

subprograms), ki je optimizirana za matrične operacije.

Model nevronske mreže se predstavi v tekstovnem formatu, ki zelo spominja na priljubljen format JSON. V vsakem objektu `layer` opišemo svoj nivo. Vsebovati mora polje `type`, ki določa vrsto nivoja, `bottom` in `top`, ki označujeta predhodni nivo in ime zbirke podatkov, kamor se shrani izhod opisanega nivoja ter ime nivoja (polje `name`). Polje `top` se potem uporabi kot vhod (polje `bottom`) v naslednji nivo. Vsi parametri so shranjeni v gnezenem objektu, ki se začne z imenom nivoja in nadaljuje s `_param`. Primer objekta za parametre konvolucijskega nivoja se imenuje `convolution_param` in v polju `num_output` vsebuje število izhodov ter v `kernel_size` velikost jedra. Na takšen način se sestavi celotna mrežo. Primer prvega konvolucijskega nivoja s pripadajočim ReLU nivojem:

```
layer {
  bottom: "data"
  top: "conv1-data"
  name: "conv1"
  type: "Convolution"
  convolution_param {
    num_output: 64
    kernel_size: 3
  }
}
layer {
  bottom: "conv1-data"
  top: "conv1-data"
  name: "conv1-relu"
  type: "ReLU"
}
```

4.3 Model konvolucijske nevronske mreže

Za model nevronske mreže smo izbrali znano šestnajst nivojsko VGG arhitekturo [5]. Takšen model se je leta 2014 izvrstno izkazal na tekmovanju ILSVRC [28], saj je bil računsko med manj zahtevnimi in je bil uvrščen med najboljše tri v večini kategorij, v nekaterih celo kot zmagovalna metoda. Uporablja več nivojev, kot je bil običaj pred izidom VGG, vendar uporablja manjše velikosti konvolucijskih oken. Rezultat manjših zaporednih konvolucij pri učenju nevronske mreže je enak eni večji, le da potrebujemo manj računanja. Dodatno se zaradi več nivojev model lahko nauči kompleksnejše vzorce z manj nevroni. Od takrat najprej je tovrsten model precej široko uporabljen za vrsto različnih problemov.

Model je sestavljen iz šestnajstih nivojev. Vhodni nivo predstavlja podatki, kjer je vhod RGB slika velikosti 224×224 slikovnih pik. Njim sledijo konvolucijski nivoji z izmenjujočimi nivoji združevanja največjih. Vsa okna konvolucijskih nivojev so velika 3×3 , velikost združevanja največjih pa 2×2 . Tako obdržimo le 25 % lokalno najboljših rezultatov. Ta se ne štejejo v skupno število nivojev. Za vsakim konvolucijskim nivojem sledi še nivo ReLU, ki skrbi za sestavljanje vzorcev. Prvotni ReLU nivoji imajo 64 izhodov, torej 64 različnih vzorcev velikost 3×3 . Naučeni vzorci so vidni na sliki 4.2.

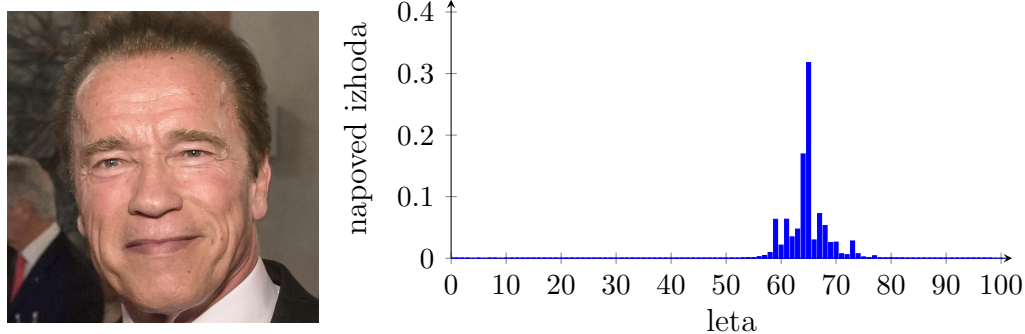


Slika 4.2: Prvih 24 vzorcev naučenih na prvem konvolucijskem nivoju.

Vizualizacija kasnejših nivojev je zaradi razbitja konvolucijskih oken precej podobna prvemu nivoju. Kasneje se število izhodnih vzorcev na konvolucijskih plasteh veča s faktorjem dva. Poveča se tudi število zaporednih nivojev ločenih s plastjo združevanja največjih, kar predstavlja večja konvolucijska okna. Na koncu sledijo trije polno povezani nivoji in izhodni nivo s 101 nevroni, ki predstavljajo leta od 0 do 100. Primer izhoda je na sliki 4.3.

Arhitektura nevronske mreže po nivojih:

Št. nivoja	Vrsta nivoja	Št. izhodov
-	vhodni nivo	$3 \times 224 \times 224$
1	konvolucijski nivo	64
2	konvolucijski nivo	64
-	združevanje največjih	
3	konvolucijski nivo	128
4	konvolucijski nivo	128
-	združevanje največjih	
5	konvolucijski nivo	256
6	konvolucijski nivo	256
7	konvolucijski nivo	256
-	združevanje največjih	
8	konvolucijski nivo	512
9	konvolucijski nivo	512
10	konvolucijski nivo	512
-	združevanje največjih	
11	konvolucijski nivo	512
12	konvolucijski nivo	512
13	konvolucijski nivo	512
14	polno povezani nivo	4096
15	polno povezani nivo	4096
16	polno povezani nivo	101 (izhod)
-	izgubni nivo (SoftmaxWithLoss)	(za učenje)



Slika 4.3: Prikaz vhodne slike (levo) iz Wikipedije, ki ni del zbirke IMDB-WIKI, in izhod nevronske mreže (desno) za vsak nevron posebej. Pravilna napoved 68, napovedana 66.

4.4 Učenje nevronske mreže

Nevronsko mrežo smo učili na grafični kartici Nvidia GeForce GTX 980 Ti. Vsebuje 2816 CUDA jeder pri frekvenci 1.2 GHz in 6 GB grafičnega pomnilnika. Računalnik ima zmogljivejši i7-6700K procesor s štirimi fizičnimi jedri in frekvenco do 4.2 GHz ter 32 GB delovnega pomnilnika. Za hitrejši dostop do podatkov je bilo vse shranjeno na SSD pogonu.

Za učenje smo preizkusili nekaj različnih vrednosti hitrosti učenja. Ta ima običajno največji vpliv na učenje in nam pove, kolikšen delež vektorja vzamemo pri gradientnem spustu, da ni prevelikih skokov ter s tem posledično nestabilnega učenja. Za izbiro vrednosti smo izhajali iz opisanih v prvotnem članku VGG [5] in te tako povečali ali zmanjšali. Hitrost učenja smo poleg referenčne vrednosti 10^{-2} preizkusili še na 10^{-1} , 5×10^{-2} , 5×10^{-3} in 10^{-3} . Pri prvi se je izkazalo, da je hitrost učenja previsoka, zato le naključno skače po prostoru. Pri drugi se je mreža že počasi učila, vendar je bilo še vedno veliko skakanja. Pri tretji in četrti vrednosti pa ravno obratno, učenje je bilo prepočasno. Referenčna vrednost 10^{-2} je bila izmed preverjenih najuspešnejša v kombinaciji z ostalimi parametri.

Za preverjanje smo odstranili 10.000 naključno izbranih slik iz zbirke IMDB-WIKI (obe združeni skupaj) za testno množico in na preostanku učili 12 ur. Po pretečenem času smo uspešnost modela preverili na testni množici z metriko najboljših 5. Torej, če je pravilen rezultat med najboljšimi petimi napovedmi se šteje za pravilno, drugače napačno.

Zmanjšali smo velikost množice za hkratno učenje (angl. batch), takšno smo tudi uporabili pri preizkušanju različnih vrednosti hitrosti učenja, iz 256 na 32 zaradi manjšega števila razredov. Prvotna VGG mreža je bila namenjena napovedovanju 1.000 različnih razredov, kar predstavlja mnogo večji prostor za učenje, kot ga imamo mi z le 101 izhodi. Velikost množice smo tako lahko zmanjšali in vseeno obdržali skoraj enako stabilno učenje. Pomemben razlog za zmanjšanje je bilo tudi pomankanje grafičnega pomnilnika. V pomnilnik gre hkrati največ 10 slik skupaj z rezultati vmesnih nivojev. Slednji zasedejo največ pomnilnika, pomembni pa so za učenje. 10 slik se je že

večkrat izkazalo za premalo število. Zato smo to množico nekoliko povečali in razbili na 4×8 slik, kar omogoča knjižnica Caffe, funkcionalnost pa ostane popolnoma enaka.

Osnovno vrednost učenja pri gradientnem spustu smo nastavili na 10^{-2} , kot so nam pokazali preizkusi. Ta se vsakih 100.000 prehodov skozi mrežo zmanjša za faktor 10, kar pripomore k stabilnejšemu učenju. Tako se hitreje približa rešitvi, v bližini pa hitrost zmanjša, da je približevanje k rešitvi natančnejše. Moment pri gradientnem spustu smo nastavili na 0,9. Ta vrednost se je že mnogokrat izkazala za najboljšo in se zato uveljavila kot standard.

Med učenjem smo v učne podatke dodali še nekaj naključnosti. Iz vhodnih slik velikosti 256×256 slikovnih pik smo na naključni lokaciji izrezali manjšo sliko velikosti 224×224 . To smo še naključno zrcalili po navpični osi in zavrteli okoli centra med -20° in 20° . S tem smo generirali nove učne primere, ki preprečujejo prekomerno prilagajanje učnim podatkom. Učenje končnega modela je za razliko od testnih, ki so bili namenjeni za izbiro parametrov namesto, 12 ur trajalo 3 dni, dokler ni doseglo konvergence.

V drugem koraku učenja smo uporabili učni del podatkovne zbirke LAP z 2.500 slikami. Z zbirko IMDB-WIKI smo model prilagodili za napovedovanje dejanske starosti, namen zbirke LAP pa je očitna starost. Naučen model smo tako z dodatnim učenjem približali napovedovanju očitne starosti. Parametri učenja so enaki kot pri prvem koraku. Zmanjšali smo korak zmanjševanja hitrosti učenja iz 100.000 na 10.000 in učili šest ur, saj je zbirka LAP manjša.

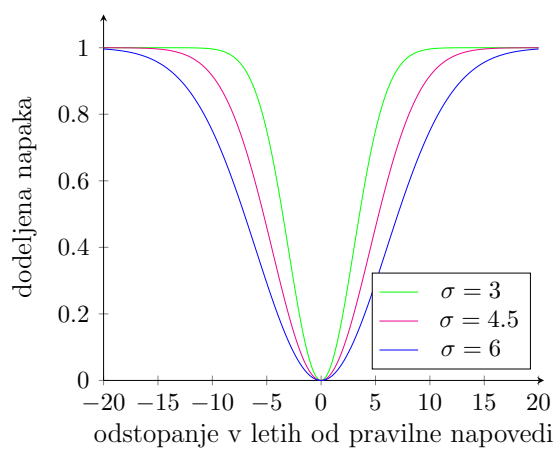
Poglavje 5

Rezultati

5.1 Rezultati za podatkovno zbirko LAP 2015

Podatkovna zbirka LAP ima funkcijo napake za vsak primer definirano po enačbi (5.1), kjer je x napovedan rezultat našega modela, μ dana povprečna ocena in σ standardni odklon ocen uporabnikov. Povprečna napaka vseh primerov je na koncu napaka modela. Vizualni prikaz enačbe je na sliki 5.1. Manjša kot je vrednost, uspešnejši je model.

$$\mathcal{E} = 1 - e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5.1)$$



Slika 5.1: Vizualni prikaz enačbe napake (5.1).

Test smo izvedli na testni množici zbirke LAP 2015 (za ocenjevalno ni objavljenih oznak primerov). Naš model je dosegel napako $0,325 \pm 0,285$. Povprečna napaka je tako 3,7 leta s standardnim odklonom $\pm 3,2$ let, v povprečju je osebe ocenil z nižjo starostjo. Napaka je bila izračunana z izraženim x iz enačbe (5.1), da se lahko primerjamo z ostalimi ekipami. Za μ in σ sta bili vzeti povprečni vrednosti iz testnega nabora. Najboljša ekipa ICT-VIPL je dosegla rezultat 0,292, kar nanese povprečno napako 3,4 leta.

Naredili smo še nekaj preizkusov občutljivosti modela glede na vhodne podatke. Nobene izmed slik nismo uporabili za učenje. Preizkusili smo občutljivost osnovnega naučenega modela glede na velikost obraza (slika 5.2) in rotacijo (slika 5.3). Pri velikosti se je pri vseh testnih primerih izkazalo, da je moral obraz na sliki predstavljati vsaj četrtnino slike, da je bila ocena kolikor toliko smiselna. Primerjava je pomembna zaradi predhodnega koraka iskanja obraza in izreza, kjer so se v skrajnih primerih pojavljali tudi največji izrezi. Z večjim poudarkom na obraz se je popravila tudi ocena. Pri rotaciji se je model izkazal za precej neobčutljivega do kota 30° . Pri večji rotaciji so bili rezultati podobno kot pri velikosti precej netočni.



Slika 5.2: Prikaz izrezov slike različnih velikosti uporabljene za preizkus modela. Ocenjena starost od največjega do najmanjšega izreza so 44, 33 in 31 let. Pravilna ocena je 33 let.



Slika 5.3: Prikaz zavrtenih vhodnih slik. Od leve proti desni so ocene 29, 28, 52 let, zavrtena pa so za 0° , 20° in 90° . Pravilna ocena 30 let.

Čas klasifikacije posamezne slike je na grafični kartici 0,5 sekunde, medtem ko na centralni procesni enoti z vsemi jedri približno 20,5 sekund. Po-hitritev je tako 41-kratna.

5.2 Primerjava s sorodnimi rešitvami

V sklopu tekmovanja LAP so bili objavljeni tudi rezultati drugih ekip [24], vendar brez podatkov o standardnem odklonu. Rezultati nad testno množico so prikazani v tabeli 5.1, kjer smo se preizkusili tudi mi. V tabeli 5.2 pa so na voljo rezultati na ocenjevalni množici, za katero nismo imeli oznak primerov. Naša metoda je tako dosegla manjšo napako kot človek in se na lestvici uvrstila med boljšo polovico.

Skoraj vse ekipe so se problema lotile z nevronskimi mrežami, le dve na klasični način z nekoliko slabšimi rezultati (Bogazici in Notts CVLab). Od nevronskih mrež sta predvsem prevladovala modela GoogleNet (mreža ima 22 nivojev) [29] in VGG, ki smo jo uporabili tudi mi. Glavna razlika med vsemi ekipami so bile predvsem podatkovne zbirke. Skoraj vse ekipe so poleg dane podatkovne zbirke uporabile tudi dodatne, saj so za učenje nevronskih mrež potrebne ogromne količine podatkov. Precej pogosti zbirki sta bili CACD [30] in MORHP [31]. Večji nabor različnih oseb kot je podatkovna zbirka imela, višje se je model uvrstil.

Zmagovalna ekipa (na ocenjevalni zbirki tekmovanja, ki je mi nismo imeli) je tako sestavila podatkovno zbirko IMDB-WIKI in jo uporabila za učenje dvajset modelov nevronske mreže. Za kontrast: mi imamo le eno. V drugem koraku so podobno kot ostale ekipe, ki so delale z nevronskimi mrežami, model še dodatno prilagodile na učni in testni množici LAP in tega uporabile za evalvacijo. Za napoved je bila vzeta povprečna vrednost vseh dvajsetih nevronske mreže. Obraze so iskali z metodo lovec na glave (angl. headhunter) [32], ki temelji na algoritmu DPM [33].

Drugouvrščena ekipa je imela le eno nevronske mreže GoogleNet. Učenje je potekalo na zunanjih podatkovnih zbirkah MORPH in še nekaj manjših, v drugem koraku pa na podatkovni zbirki LAP. Obraze so iskali s pomočjo spletne storitve Face++ [34] in Microsoftovih kognitivnih storitev [35].

Tretja ekipa je naučila več nevronske mreže na zbirkah MORPH, CACD in FGNet. Te so razvrstile primere v 10 različnih skupin, nato so z naključnimi gozdovi (angl. random forest) [36] in SVM [37] za regresijo določili končni izhod.

Mesto	Ekipa	Napaka	Model	Napovedovanje
1	ICT-VIPL	0,292297	GoogleNet (CNN)	klasifikacija
2	CVL-ETHZ	0,295116	VGG (CNN)	klasifikacija
3	WVU CVL	0,316289	GoogleNet (CNN)	regresija
-	Naša metoda	0,325119	VGG (CNN)	klasifikacija
4	AgeSeer	0,327321	VGG (CNN)	regresija
-	Človek	0,34	-	-
5	Enjuto	0,370656	CNN	regresija
6	SEU-NJU	0,380615	VGG (CNN)	ni podatka
7	Lab219A	0,477079	CNN	ni podatka
8	Bogazici	0,483337	računanje značilk	regresija
-	UMD	-	CNN	klasifikacija, nato regresija
-	Sungbin Choi	-	GoogleNet (CNN)	klasifikacija
-	Notts CVLab	-	računanje značilk	regresija

Tabela 5.1: Prikaz rezultatov ekip na tekmovanju LAP na testni množici.

Mesto	Ekipa	Napaka
1	CVL_ETHZ	0,264975
2	ICT-VIPL	0,270685
3	WVU_CVL	0,294835
4	SEU-NJU	0,305763
5	UMD	0,373352
6	Enjuto	0,374390
7	Sungbin Choi	0,420554
8	Lab219A	0,499181
9	Bogazici	0,524055
10	Notts CVLab	0,594248

Tabela 5.2: Prikaz rezultatov ekip na tekmovanju LAP na ocenjevalni množici.

Poglavje 6

Zaključek

Konvolucijske nevronske mreže smo preizkusili na problemu napovedovanja starosti osebe zgolj s fotografije obraza. V prvem delu diplomskega dela smo predstavili osnovno teorijo nevronskih mrež, kjer smo opisali osnovno zgradbo, gradnike, nekaj najbolj pogosto uporabljenih nivojev in najširše uporabljeno metodo učenja z gradientnim spustom. V nadaljevanju smo opisali obsežno podatkovno zbirko IMDB-WIKI, ki vsebuje ogromno fotografij predvsem znanih oseb z označeno starostjo in precej manjšo podatkovno zbirko LAP, ki je dana v sklopu tekmovanja in na kateri smo tudi preizkusili našo metodo. Sledil je praktični del diplomske naloge, ki je v prvem koraku zajemal pripravo podatkov. V danih podatkovnih zbirkah smo našli vse obraze z novo metodo NPD in izvedli filtriranje vseh učnih primerov. Šestnajstnivojsko nevronske mrežo smo naučili v široko uporabljeni knjižnici Caffe za delo s konvolucijskimi nevronskimi mrežami, na koncu pa naredili evalvacijo rezultatov. Dobljene rezultate smo primerjali na podatkovni zbirki LAP.

Naša metoda se je kljub nekoliko preprostejšemu modelu in pripravi podatkov izkazala za precej uspešno, saj smo imeli manjšo napako kot človek in se zelo približali najboljšim trem ekipam. Ugotovili smo, da je bolj kot metoda pomemben nabor podatkov. Več različnih primerov kot imamo na voljo, bolje se bo model odrezal.

Prostora za nadgradnjo je še kar nekaj. Ena izmed izboljšav, ki bi verjetno precej pripomogla k večji uspešnosti, je vključitev dodatnih podatkovnih zbirk. Druga, verjetno najbolj obetavna pa bi bila kompleksnejši, petdeset ali več nivojski model nevronske mreže, naučen z metodo učenja z ostanki. To metodo smo omenili v prvem poglavju, vendar je nismo uporabili zaradi izjemno dolgih časov učenja. Izkazala se je za izjemno uspešno. Predvidevamo, da bo v prihodnjih letih VGG zamenjal ResNet in novejša arhitekture polno povezanih nevronskih mrež [38].

Ravno čas učenja in splošna računska zahtevnost nam je predstavljala največjo težavo. Nevronske mreže imajo veliko število parametrov za učenje, kjer lahko že majhna sprememba vpliva na uspešnost modela. Večje število različnih parametrov pa ravno zaradi časa učenja težko preizkusimo. Visoka računska zahtevnost nam ravno tako omeji aplikativno rabo, kot je npr. določanje starosti z videa v realnem času ali pa uporaba algoritma v mobilni aplikaciji.

Literatura

- [1] Looking at people faces (LAP). <http://gesture.chalearn.org>. Dostopano: 17. julij 2016.
- [2] Y. Fu, G. Guo, and T. S. Huang, “Age synthesis and estimation via faces: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 1955–1976, 2010.
- [3] K. Ramesha, N. Srikanth, K. B. Raja, R. K. Venugopal, and L. M. Patnaik, “Advanced biometric identification on face, gender and age recognition,” in *Advances in Recent Technologies in Communication and Computing*, vol. 29, pp. 23–27, Oct. 2009.
- [4] X. Geng, Z. H. Zhou, and K. Smith-Miles, “Automatic age estimation based on facial aging patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2234–2240, 2007.
- [5] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv Computing Research Repository*, vol. abs/1409.1556, Sept. 2014. (<https://arxiv.org/abs/1409.1556>).
- [6] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *The IEEE International Conference on Computer Vision*, pp. 1026–1034, Dec. 2015.
- [7] A. Zell, *Neuronaler Netze*. Addison-Wesley, 1994.

-
- [8] R. Lippmann, “An introduction to computing with neural nets,” *IEEE ASSP Magazine*, vol. 4, no. 2, pp. 4–22, 1987.
 - [9] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Unsupervised learning of hierarchical representations with convolutional deep belief networks,” *Communications of the ACM*, vol. 54, no. 10, pp. 95–103, 2011.
 - [10] Y. Freund and R. E. Schapire, “Large margin classification using the perceptron algorithm,” *Machine Learning*, vol. 37, no. 3, pp. 277–296, 1999.
 - [11] K.-I. Funahashi, “On the approximate realization of continuous mappings by neural networks,” *Neural networks*, vol. 2, no. 3, pp. 183–192, 1989.
 - [12] S. Geman, E. Bienenstock, and R. Doursat, “Neural networks and the bias/variance dilemma,” *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.
 - [13] D. E. Rumelhart, G. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, 1986.
 - [14] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *IEEE International Conference on Evolutionary Computation*, pp. 69–73, May 1998.
 - [15] D. J. Montana and L. Davis, “Training feedforward neural networks using genetic algorithms,” in *IEEE International Joint Conference on Artificial Intelligence*, vol. 89, pp. 762–767, Aug. 1989.
 - [16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Jun. 2016.
 - [17] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, “On the number of linear regions of deep neural networks,” in *Advances in neural infor-*

- mation processing systems*, pp. 2924–2932, Curran Associates, Inc., Feb. 2014.
- [18] G. Strang and T. Nguyen, *Wavelets and filter banks*. SIAM, 1996.
- [19] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv Computing Research Repository*, vol. abs/1207.0580, 2012. (<http://arxiv.org/abs/1207.0580>).
- [20] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *IEEE International Conference on Machine Learning*, pp. 807–814, 2010.
- [21] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten, “Neural-gas’ network for vector quantization and its application to time-series prediction,” *IEEE Transactions on Neural Networks*, vol. 4, pp. 558–569, Jul. 1993.
- [22] Podatkovna zbirka IMDB-WIKI. <https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/>. Dostopano: 17. julij 2016.
- [23] Podatkovna zbirka LAP 2016. <https://competitions.codalab.org/competitions/7511>. Dostopano: 17. julij 2016.
- [24] S. Escalera, J. Fabian, P. Pardo, X. Baró, J. González, H. J. Escalante, D. Mišević, U. Steiner, and I. Guyon, “Chalearn looking at people 2015: Apparent age and cultural event recognition datasets and results,” in *IEEE International Conference on Computer Vision Workshop*, pp. 243–251, Dec. 2015.
- [25] S. Liao, A. K. Jain, and S. Z. Li, “A fast and accurate unconstrained face detector,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 211–223, 2016.

- [26] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof, “Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization,” in *IEEE International Conference on Computer Vision Workshops*, pp. 2144–2151, Nov. 2011.
- [27] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” *arXiv Computing Research Repository*, vol. abs/1408.5093, Jun. 2014. <http://arxiv.org/abs/1408.5093>.
- [28] Tekmovanje ILSVRC. <http://www.image-net.org/challenges/LSVRC/>. Dostopano: 28. junij 2016.
- [29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Jun. 2015.
- [30] B.-C. Chen, C.-S. Chen, and W. H. Hsu, “Cross-age reference coding for age-invariant face recognition and retrieval,” in *European Conference on Computer Vision*, pp. 768–783, Sept. 2014.
- [31] K. Ricanek Jr. and T. Tesafaye, “Morph: A longitudinal image database of normal adult age-progression,” in *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 341–345, Apr. 2006.
- [32] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, “Face detection without bells and whistles,” in *European Conference on Computer Vision*, pp. 720–735, Springer, Sept. 2014.
- [33] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.
- [34] Face++. <http://www.faceplusplus.com>. Dostopano: 17. julij 2016.

-
- [35] Microsoft cognitive services. <http://https://www.microsoft.com/cognitive-services>. Dostopano: 17. julij 2016.
 - [36] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.
 - [37] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
 - [38] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, Jun. 2015.